

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ

УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ЗНАКОМСТВО С РЕЛЯЦИОННЫМ ПРОЕКТИРОВАНИЕМ ДАННЫХ

Отчет по индивидуальному заданию по дисциплине «Основы разработки баз  
данных»

Студент гр. 573-3

\_\_\_\_\_ Р.В. Слинков

" \_\_\_\_ " \_\_\_\_\_ 2025 г.

Преподаватель

\_\_\_\_\_ Р.О. Остапенко

" \_\_\_\_ " \_\_\_\_\_ 2025 г.

Томск 2025

## **Введение**

В условиях динамичного развития логистики и торговли эффективное управление складскими запасами становится критически важным для бизнеса. Современные системы складского учета требуют не только автоматизации процессов, но и надежного хранения, обработки и анализа больших объемов данных. Разработка структурированной базы данных является ключевым этапом в создании такой системы, так как она обеспечивает целостность данных, минимизирует дублирование информации и позволяет оперативно получать актуальные сведения о состоянии товарных запасов.

Цель данной работы — разработать концептуальную и реляционную модель базы данных для системы "Складской учет".

Результатом станет нормализованная реляционная модель, гарантирующая непротиворечивость данных, минимизацию избыточности и удобство выполнения запросов для анализа складских операций.

### **Задачи:**

1. Определить отношения (таблицы) и связи между отношениями в предметной области на основе концептуальной информационной модели из индивидуального задания №1.
2. Определить атрибуты, а так же первичные и внешние ключи в отношениях (таблицах).
3. Подвергнуть полученные отношения процессу нормализации (использовать только первые три нормальные формы).
4. Полученный после нормализации результат представить в виде реляционной модели данных в методологии IDEF1x.

# 1 ОСНОВНАЯ ЧАСТЬ

## 1.1 Определение отношений в таблицах для построения модели

Для построения модели данных (рисунок 1.1), на основе концептуальной модели из прошлого задания, мы можем определить следующие сущности и связи между ними.

### 1. Товар:

- Наименование товара;
- Тип товара;
- Срок годности товара.

Связи:

- Один товар может участвовать в нескольких поставках (1:M → Поставка).

### 2. Учет товара:

- Дата поступления;
- Количество;
- Дата последнего перемещения;
- Поставка;
- Поставщик;
- Зона хранения;
- Сотрудник.

Связи:

- В каждом учете может храниться несколько поставщиков и каждый поставщик может поставлять множество товаров (M:M → Поставщик);
- Каждый учет ведется одним сотрудником, но каждый сотрудник может вести множество учетов (M:1 → Персонал);
- Каждый учет товара связан с одной зоной хранения, но одна зона хранения может иметь несколько учетов товаров (M:1 → Зона хранения).

### 3. Поставщик:

- Наименование компании;
- Контактное лицо;
- Телефон;
- Адрес.

Связи:

- В каждом учете может храниться несколько поставщиков и каждый поставщик может поставлять множество товаров ( $M:M \rightarrow$  Учет товаров);

- Один поставщик может поставлять несколько различных товаров в рамках разных поставок ( $1:M \rightarrow$  Поставка).

#### 4. Персонал:

- ФИО;
- Должность;
- Фотография;
- Номер телефона.

Связи:

- Каждый учет ведется одним сотрудником, но каждый сотрудник может вести множество учетов ( $1:M \rightarrow$  Учет товаров).

#### 5. Зона хранения:

- Название зоны;
- Склад;
- Тип зоны;
- Вместимость.

Связи:

- Каждый учет товара связан с одной зоной хранения, но одна зона хранения может иметь несколько учетов товаров ( $1:M \rightarrow$  Учет товара);

- Каждая зона находится на одном складе и на одном складе может быть несколько зон хранения ( $M:1 \rightarrow$  Склад).

#### 6. Склад:

- Название склада;
- Адрес.

Связи:

- Каждая зона находится на одном складе и на одном складе может быть несколько зон хранения (1:M → Зона хранения).

#### 7. Поставка:

- Товар;
- Поставщик;
- Дата поставки;
- Количество поставленного товара.

Связи:

- Один поставщик может поставлять несколько различных товаров в рамках разных поставок (M:1 → Поставщик);
- Одна поставка может быть связана с несколькими учетами товара (1:M → Учет товара).

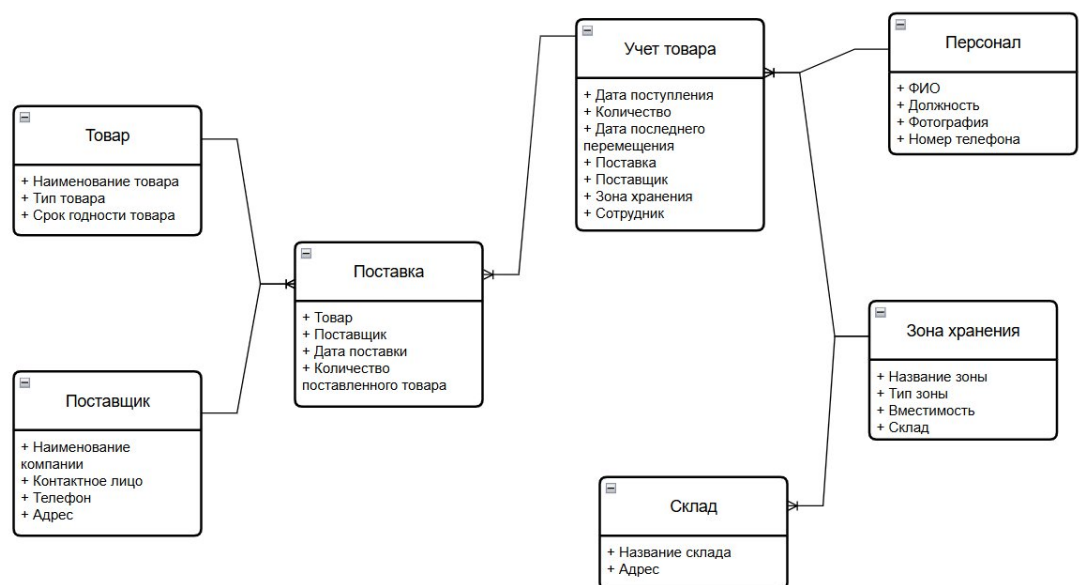


Рисунок 1.1 – Исходная модель данных

## 1.2 Выделение первичных и внешних ключей

Для представления данных в реляционной модели, необходимо выразить первичные и внешние ключи:

### 1. Товар:

- РК: product\_id (Товар);
- Наименование товара;
- Тип товара;
- Срок годности товара.

### 2. Учет товара:

- РК: productAcc\_id (Учет товара);
- FK: supply\_id (Поставка);
- FK: employee\_id (Сотрудник);
- FK: storage\_id (Зона хранения);
- Дата поступления;
- Количество;
- Дата последнего перемещения.

### 3. Персонал:

- РК: employee\_id (Сотрудник);
- ФИО;
- Фотография;
- Должность;
- Номер телефона.

### 4. Поставщик:

- РК: supplier\_id (Поставщик);
- Наименование компании;
- Контактное лицо;
- Телефон;
- Адрес.

### 5. Зона хранения:

- PK: storage\_id (Зона хранения);
- FK: warehouse\_id (Склад);
- Вместимость;
- Тип зоны;
- Название зоны.

6. Склад:

- PK: warehouse\_id (Склад);
- Название;
- Адрес.

7. Поставка:

- PK: supply\_id (Поставка);
- FK: product\_id (Товар);
- FK: supplier\_id (Поставщик);
- Дата поставки;
- Количество поставленного продукта.

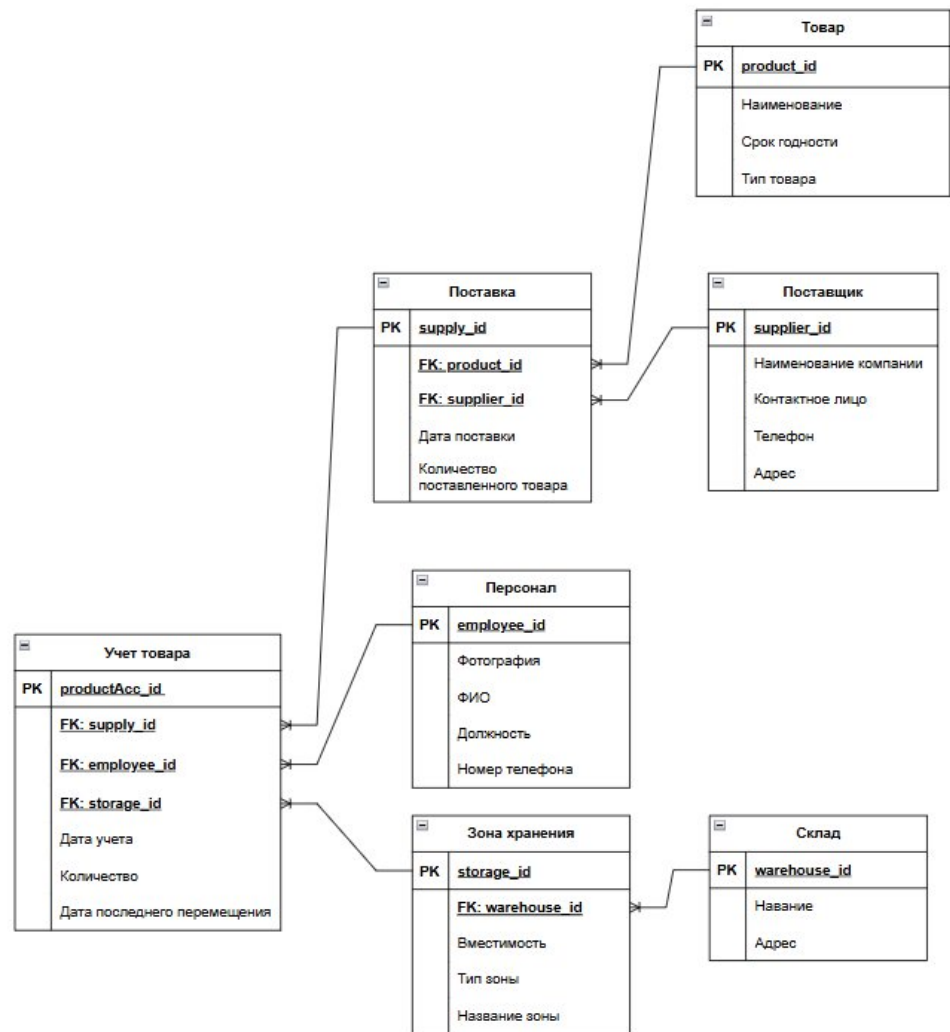


Рисунок 1.2 – Исходная модель данных в методологии IDEF1x



### 1.3 Нормализация

Нормализация – процесс реорганизации данных путем ликвидации повторяющихся групп и иных противоречий в хранении данных с целью приведения таблиц к виду, позволяющему осуществлять непротиворечивое и корректное редактирование данных.

Чтобы нормализовать предложенные данные по первой нормальной форме (1NF), нужно выполнить несколько шагов. Первая нормальная форма требует, чтобы все атрибуты таблицы содержали только атомарные значения, то есть каждый столбец должен содержать одно значение (не массив или список).

Исходя из анализа модели, можно сказать, что она соответствует 1NF, так как все атрибуты атомарны.

Чтобы привести данные ко второй нормальной форме (2NF), необходимо выполнить два основных шага:

- Удалить частичные зависимости, т.е. все атрибуты должны зависеть от всего первичного ключа, а не только от его части.
- Обеспечить, чтобы каждая таблица имела только одну зависимость от ключа. Если атрибут зависит от части составного ключа (если он существует), его нужно перенести в другую таблицу.

Анализ модели в соответствии с 2NF.

1. Поставщик:

- Все атрибуты зависят от полного первичного ключа (supplier\_id);
- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

2. Товар:

- Все атрибуты полностью зависят от первичного ключа (product\_id);
- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

### 3. Персонал:

- Все атрибуты полностью зависят от первичного ключа (employee\_id);

- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

### 4. Зона хранения:

- Все атрибуты полностью зависят от первичного ключа (storage\_id);

- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

### 5. Склад:

- Все атрибуты полностью зависят от первичного ключа (warehouse\_id);

- Storage\_id не влияет на неключевые атрибуты;

- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

### 6. Учет товара:

- Все атрибуты полностью зависят от первичного ключа (productAcc\_id);

- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

### 7. Поставка:

- Все атрибуты полностью зависят от первичного ключа (supply\_id);

- Не имеет частичных зависимостей.

Вывод: Соответствует 2NF.

Чтобы привести данные к третьей нормальной форме (3NF), нужно выполнить два шага:

- Удалить транзитивные зависимости, т.е. атрибуты, которые зависят от других атрибутов, а не от первичного ключа.

- Обеспечить, чтобы все атрибуты в таблице зависели напрямую от первичного ключа и не имели зависимости от других неключевых атрибутов.

В нашем случае, мы будем искать атрибуты, которые зависят от других атрибутов, и переработаем структуру, чтобы устранить транзитивные зависимости.

В процессе преобразования во вторую нормальную форму, мы автоматически перешли в третью, поскольку все транзитивные зависимости перешли в другие таблицы.

По итогам трёх форм нормализации мы можем построить реляционную модель данных в методологии IDEF1x (рисунок 1.3).

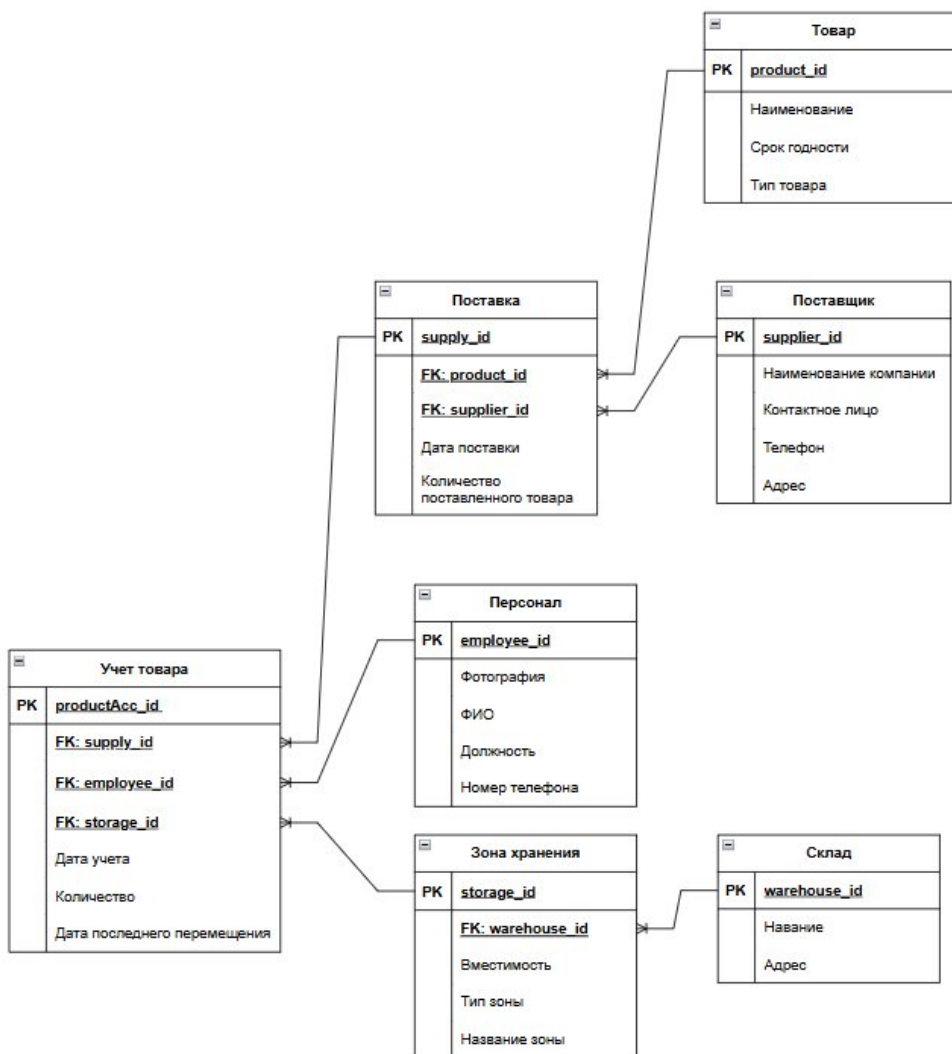


Рисунок 1.3 – Реляционная модель данных в методологии IDEF1x

## **Заключение**

В ходе работы была разработана модель базы данных для автоматизированной системы управления складскими процессами. Определены основные сущности, включая поставщиков, товары, персонал, зоны хранения, склады, учёт товаров и поставки, с указанием их атрибутов и взаимосвязей. Чётко выделены первичные и внешние ключи для всех сущностей, что обеспечило логическую целостность и связанность данных.

Для улучшения структуры базы данных был применен процесс нормализации до третьей нормальной формы (3NF). Это позволило устранить избыточность данных, минимизировать аномалии при обновлении и повысить целостность информации.